SECURITY CLASSIFICATION OF THIS PAGE

(2)

REPORT DOCUMENTATION PAGE

DTIC FILE COPY

AD-A209 630

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 89-0775
6a. NAME OF PERFORMING ORGANIZATION State University of New York Dept of Mathematics 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFOSR
6c. ADDRESS (City, State, and ZIP Code) Binghamton, NY 13901	7b. ADDRESS (City, State, and ZIP Code) BLDG 410 BAFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR 75-2813
8c. ADDRESS (City, State, and ZIP Code) BLDG 410 BAFB DC 20332-6448	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102F PROJECT NO. 2304 TASK NO. A5 WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

DISCRETE TIME ANALYSIS OF A SHUT DOWN QUEUEING SYSTEMS

12. PERSONAL AUTHOR(S)

Eugene M. Klimko

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM ______ TO ______	1977	35

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

DTIC ELECTE JUL 1989 E

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) 767-5025 22c. OFFICE SYMBOL NP

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

89 ... 123

AFOSR-TR- 89-0775

DISCRETE TIME ANALYSIS OF A SHUT DOWN QUEUEING SYSTEM

by

Eugene M. Klimko

QUALITY INSPECTED 2

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Research sponsored by the Air Force Office of Scientific Research under Grant No. AFOSR 75-2813. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

PREFACE

This report contains the results of the studies supported by contract number AFOSR 75-2813 with the Air Force Office of Scientific Research for the period April 1, 1975 to June 30, 1976. The principle investigator for this project was Dr. Eugene M. Klimko of the Mathematical Sciences Department of the State University of New York at Binghamton. The only personnel supported by this contract was the principal investigator during the period July - August 1975.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Analysis of a Shutdown Queueing System.

1. Introduction. The purpose of this study is to analyze a special type of job shop queueing system which has the following features.

(i) A finite number of customers is present initially and no new customers arrive;

(ii) there are two service stations, each with its own input and each serving the output of the other; and

(iii) as soon as a customer is served by both stations, he leaves the system.

A queueing system with these features is called a shutdown queueing system and it may also be called a clearance problem.

Queueing problems of this type arise in connection with fuel and supply facilities for aircraft. A fixed number of aircraft are scheduled for fueling and supply loading.

Each of these operations is conducted at two separate stations called station I and station II. Initially, station I (which may be the fuel station) has M customers while station II (supply station) has N customers. After an aircraft, henceforth called a customer, initially at station I completes its service, it enters the queue for service at station II. When its service at station II is also complete, it leaves the system. The service time at both stations is assumed to be a random quantity. Similarly, a customer initially in the station II queue, moves to the station I queue and leaves the system after his service is complete.

The length of time required for the entire operation, or the clearance of the queue, is a random quantity whose probability distribution is of major interest. Also of interest is the idle time distribution at each of the supply stations. Since no new customers join the queue, only nonsteady state results are of interest.

Under the assumption that the service time distribution is negative exponential, this problem was studied by Milch and Waggoner
They obtained Laplace transforms of the probability distribution of the total operational time, the idle time distribution for each station and the waiting time distribution for various customers in the system. These transforms were quite complex and not easily inverted.

The work of Milch and Waggoner can be studied in greater detail and generality using discrete time methods rather than continuous time models. Discretizing the time variable makes it possible to retain all conditional probability distributions for the entire time interval of the operation of both stations. Because of the many possible interactions between the two queues, the analysis of this general model by classical analytical methods is an intractable problem. The analysis of Milch and Waggoner relies heavily on the memoryless property of the exponential distribution.

Even using discrete time, the numberical solution of this problem requires extensive computing. Many realizations of the process occur with negligible probability, which may be conveniently discarded early in the complex computational algorithm reported on here. On the other hand, it is possible to obtain upper bounds on the probability neglected at each stage.

2. Statement of the Problem. Consider two stations, I and II. Initially, station I has N customers and station II has N customers. The service time at each station is measured in discrete time units. The probability that a customer at station I requires i units of service time is X_i for $i = 1, 2, \ldots, L_1$ while at station II a customer requires j units of service with probability Y_j for $j = 1, 2, \ldots, L_2$. The two stations operate independently. After a customer completes service at the station of his initial assignment, he joins the end of the queue of the other station. The amount of transit time is assumed to be zero. (This assumption is not restrictive, since the transit time could be counted as part of the service time.)

After being served by both stations, the customer leaves the system. It is possible for the queue at either station to become empty. The station then becomes idle and remains so until a customer completes service at the other station. Such idleness may repeatedly occur during the service operation of the system.

3. Random Walk Representation. The progress of this queueing system may be represented by a random walk process on a two dimensional lattice. The random walk starts at time zero at the origin. Thereafter, the system is

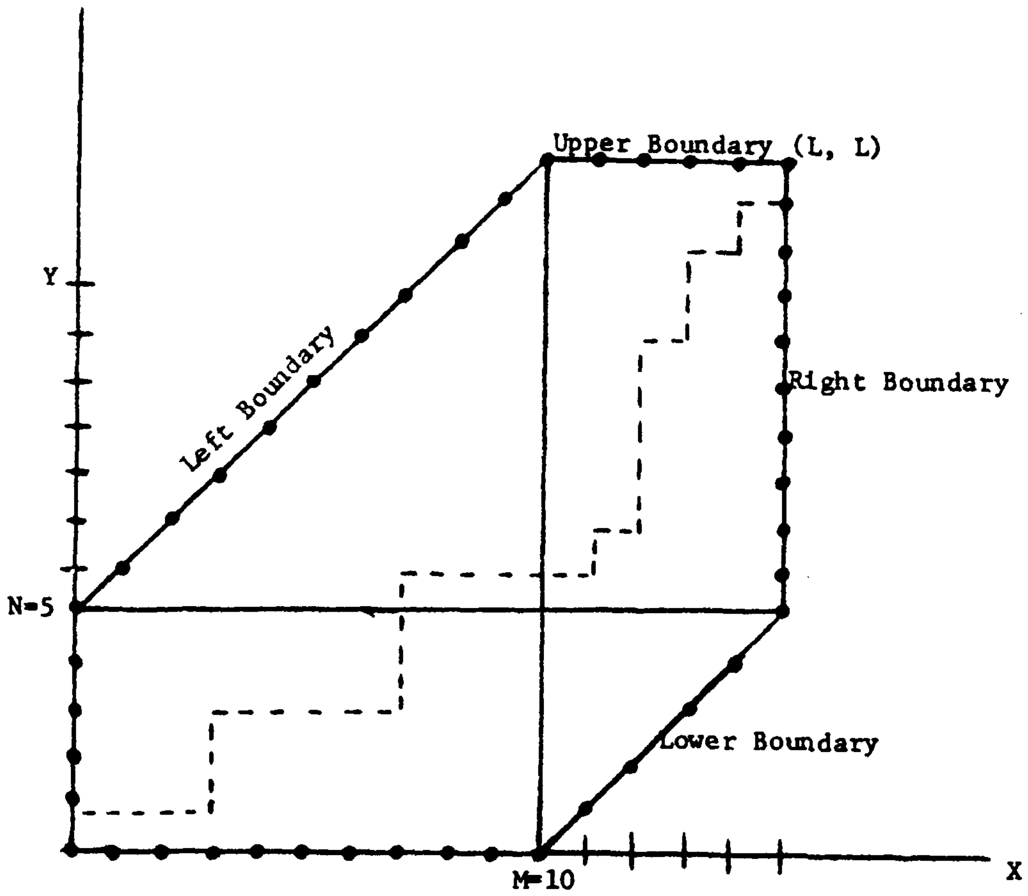

Figure 1   Random Walk Representation

represented by the pair, (ν, σ) where ν is the total number of customers served by station I and σ is the total number of customers served by station II. Each time a customer completes service at either station, a transition takes place to state $(\nu + 1, \sigma)$ if the service completion occurred at station I or to the state $(\nu, \sigma + 1)$ if the completion occurred at station II. The state $(0, 0)$ is the starting point of the operation of the queue and the state (L, L) where $L = M + N$ is the termination point of the operation, since all of the customers have been served at this time. Each realization of the queueing process is a path from $(0, 0)$ to (L, L) but not every possible path is a possible sequence of service operations by our queueing system. If $X(t)$ and $Y(t)$ represent the number of customers served at time t by stations I and II respectively, then the rules of operation dictate that

$$X(t) \leq M + Y(t) \qquad (1)$$
$$Y(t) \leq N + X(t)$$

These conditions determine boundary conditions on the paths from $(0, 0)$ to (L, L) which may be bona fide realizations of our queueing system. Figure 1 shows the boundaries imposed on the lattice as well as a typical path representing a realization of the queueing system. The realizations are constrained to lie within an irregular hexagon H determined by the boundary lines $X = 0$, $Y = 0$, the left boundary $Y = X + N$, the upper boundary $Y = M + N$, the right boundary $X = M + N$ and the lower boundary $Y = X - M$.

Whenever a customer finishes service at station I, the path moves to the right while a completion at station II causes the path to move upward. Whenever a path touches either the left boundary or the lower boundary, the path must move to the right or upward respectively.

4. Computational Methods for the Total Operation Time Distribution. As pointed out above, each path from the origin $(0, 0)$ to the terminal point (L, L), which lies wholly within the hexagon H represents an operation of the queueing system. Also associated with each path is a probability distribution on the length of time of operation of the queue given that the particular path was traversed. With this in mind, we may describe the state of the system by means of a quintuple

(2) $$(h, j, \nu, \tau, t)$$

where h and j are the x, y coordinates of the lattice point the system currently occupies, and are the number of units of service time given to the customers at stations I and II respectively and t is the total time elapsed.

We next define the various first passage probabilities to a lattice point. Suppose that the process is initially in state (h_0, j_0) with service to the customer in station I just beginning and the customer at station II having units of service time remaining to completion of his service. Assume that the system arrives for the first time in state (h, j) and that all paths from (h_0, j_0) to (h, j) are permissible; i.e., be within the hexagon H. We define the quantity

(3) $$\psi_1(h_0, j_0, h, j, \nu, \tau, t),\ t = 0, 1,$$

to be the probability that t units of time are required for a passage from (h_0, j_0) to (h, j) where the passage is from the right with ν remaining units of service required for the customer at station I initially and τ units of time remaining for the customer at station II when the passage to state (h, j) first occurs. Five additional probabilities similar to ψ, are needed the quantity

(4) $$\phi_2(h_0, j_0, h, j, \nu, \tau, t)$$

is similar to ψ_1 except that ν represents the number of remaining units of service time required for the customer at station II rather than station I when the state of the system is (h_0, j_0). The quantities ψ_1 and ψ_2 are probabilities of transition into a state where the transition is to the right into the final state (h, j) rather than from below. That is, a customer at station I has just completed his service while the customer at station II has τ units of service time remaining.

Analogous to the ψ_1 and ψ_2, we define

(5) $$\phi_1(h_0, j_0, h, j, \nu, \tau, t)$$

to be the probability that the system makes a transition from state (h_0, j_0) to the state (h, j) from below in t units of time with τ units of service time remaining for the customer at station I and initially ν units of time remained for for the customer at station II in the initial state (h_0, j_0). Also the quantity

(6) $$\phi_2(h_0, j_0, h, j, \nu, \tau, t)$$

is the probability of transition from (h_0, j_0) to (h, j) in t units of time with τ service units remaining for the customer at station I and initially ν units of service time remained for the customer at station I rather than station II.

In addition to the transitions from the right $(h - 1, j) \rightarrow (h, j)$ and the transition from below $(h, j - 1) \rightarrow (h, j)$, it is possible to have a diagonal transition of the form $(h - 1, j - 1) \rightarrow (h, j)$. This corresponds to two customers one at each station both finishing service simultaneously. For this situation, we need two additional probabilities:

(7) $$\theta_1(h_0, j_0, h, j, \nu, t),$$

and

(8) $$\theta_2(h_0, j_0, h, j, \nu, t).$$

These are the probabilities that a diagonal transition from state (h_0, j_0) to state (h, j) takes place in t units of time. Additionally, ν units of service time remained fro the customer at station I (θ_1) respectively station II (θ_2) in the initial state (h_0, j_0).

Formulas for these transition probabilities are given below.

The computation of the total service time distribution proceeds in four separate stages. The first three stages consist of finding the probability distribution on the time of first passage to various boundaries in the hexagon Π. The last stage consists of finding the passage probability to the terminal state (L, L).

The phase I boundary consists of the two line segment joining the points (0, N), (M, N) and (M, 0). It should be obvious that the passage from the origin (0, 0) to this boundary occurs with probability one. The probability distributions at lattice points along this boundary are given by the quantities

$$\psi_1(0, 0, M, j, 0, \tau, t) \tag{9}$$

where $j = 0, \ldots, N$; $\tau = 1, \ldots, L_2$; and $t = 0, 1, \ldots, T$ where T is the maximum possible total service time required to reach this boundary. These correspond to the boundary on the line segment from (M, 0) to (M, N). Along the segment from (0, N) to (M, N) the probabilities are given by

$$\phi_1(0, 0, h, N, 0, \tau, t) \tag{10}$$

where $h = 0, \ldots, M$, $\tau = 1, \ldots, L_1$ and $t = 0, 1, \ldots T$. The corner probability is given by

$$\theta_1(0, 0, M, N, 0, t). \tag{11}$$

Note that since $\nu = 0$ in each case the subscript 1 could be replaced by a 2.

The phase II boundary is determined by the line segments connecting the points (M, L), (M, N) and (N, L). The computation of the first passage distribution to these boundaries is considerably more complex than the probabilities for the phase I boundary.

The lattice points along the phase I boundary, constitute a partition of the probability space. Therefore, by the law of total probability, the distribution along the phase II boundary may be obtained by computing the distribution given that the process started at a point along the phase I boundary and summing over the phase I boundary. That is

$$P[(h, j, \tau, t, s)] = \sum P[(h, j, \tau, t, s) | (h_0, j_0, \tau_0, t_0, s_0)] \, P[(h_0, j_0, \tau_0, t_0, s_0)] \tag{12}$$

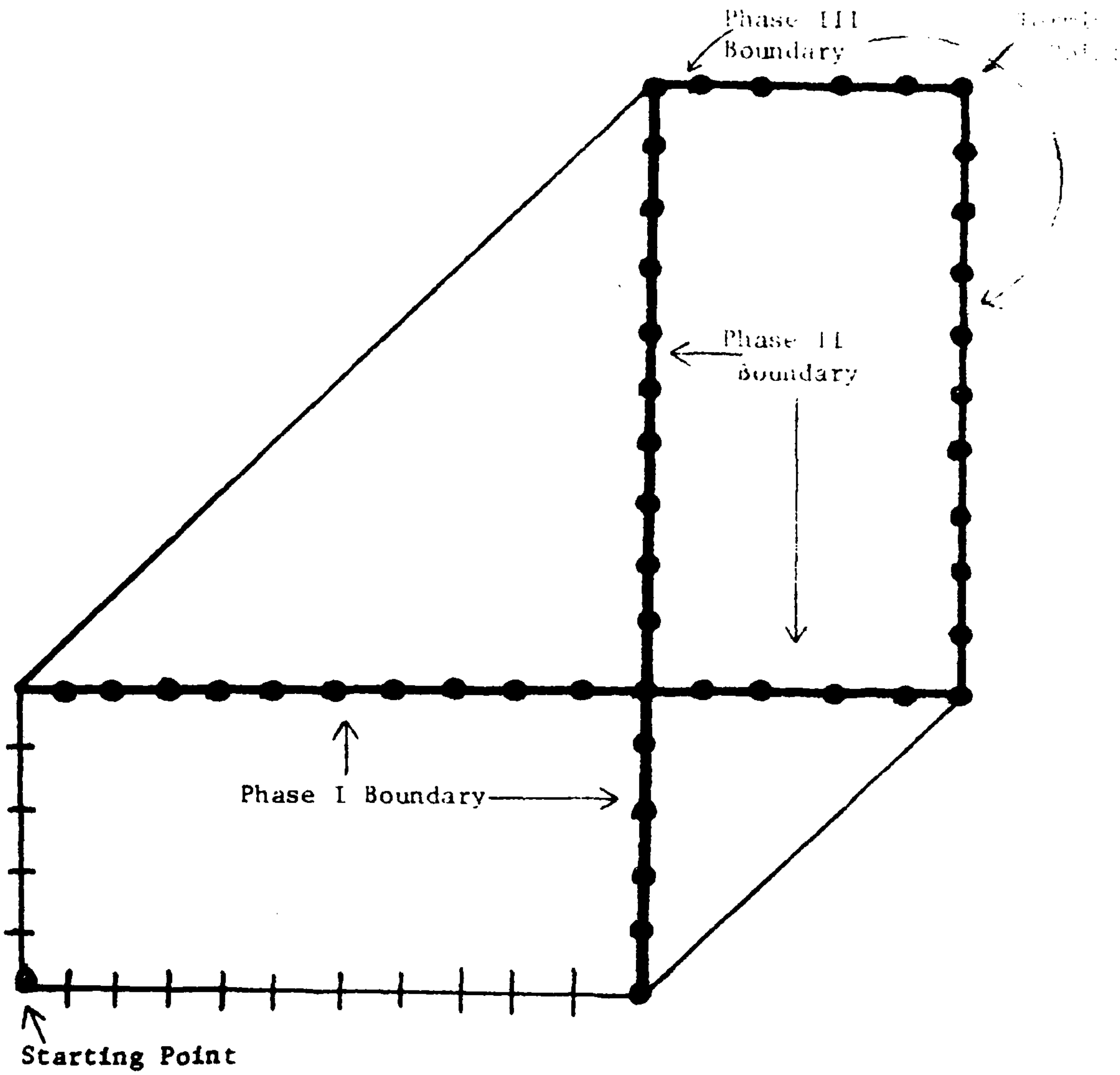

Figure 2 Boundaries for Processing Stages

where (h, j, τ, t, s) is a state along the phase II boundary. (h, j) are the lattice points. τ is the service time remaining for either customer, $s = 1$ or 2 denoting the customer for which there is residual service time and t is the total operational time up to this point. The state $(h_0, j_0, \tau_0, t_0, s_0)$ is a state along the phase I boundary with the quantities h_0, j_0, τ_0, t_0, s_0 having the same meaning as described for the phase II boundary. The summation is taken over all possible values of $(h_0, j_0, \tau_0, t_0, s_0)$. That is, for $h_0 = M$, $j_0 = 0, 1, \ldots, N - 1$, $\tau_0 = 1, 2, \ldots, L_2$, $s_0 = 2$ and for $j_0 = N$, $h_0 = 0, \ldots, m - 1$, $\tau_0 = 1, 2, \ldots, l_1$, $s_0 = 1$. Further details of this computation are given in the section on computation.

The phase III computation is similar in strategy to the phase II strategy. The phase III boundary consists of the two line segments connecting the points (M, L), (L, L) and (L, N). The strategy is to regard the phase II boundary as a partition of the probability space and to compute the conditional boundary distribution along the phase III boundary given that the process passes through the particular point along the phase II boundary.

The phase IV computation consists of finding the probability distribution on the time to enter the state (L, L) given that the process passes through one of the lattice points on the phase III boundary.

Each lattice point (h, j) along the phase I boundary generates a distribution along the phase II boundary. The total of such distributions is $M + N + 1$. Each lattice point of this distribution generates a distribution along the phase III boundary. Thus, each lattice point along the phase I boundary generates $(M + N + 1)(M + N - 1)$ distributions along the phase III boundary. These distributions are joint distributions in τ and t. Many of these distributions have a total probability which is quite small. Thus, if at the first phase of processing, some distributions are eliminated because the passage probability at that point is small, the elimination will have a negligible effect on the accuracy of the total operational time distribution, but will eliminate the computation of many probability distributions which will save a considerable amount of computer time.

The large number of distributions involved throughout the computation also creates a need for a substantial amount of storage for these distributions. To solve this problem, auxiliary disk storage is used to store the distributions along the particular boundaries for each phase of the processing. An indexing algorithm was developed to keep tract of the bookkeeping involved in this process.

A check on computational accuracy and correctness is made at each stage of the processing by using the law of total probability on each of the boundaries for phase I, phase II and phase III. In addition, certain internal auditing schemes are used to check for accuracy and correctness of the algorithm.

5. Computational Formulas for ψ, ϕ and θ. We now consider the problem of computing the passage probabilities from a given initial lattice point (h_0, j_0) to another lattice point (h, j). It is assumed that the process remains within the square whose vertices are (h_0, j_0), (h_0, j), (h, j_0) and (h, j), but may otherwise follow any path from (h_0, j_0) to (h, j). We consider the case of ψ_1 first. THe passage into the state (h, j) is from the right. In addition, ν units of time remained to service the customer at station I. A transition of this type takes place in t units of time if and only if $j - j_0$ customers have been served at station II and service has begun for an additional customer who requires τ additional units of service to complete his processing. Simultaneously, the customer at station I must complete his ν units of service time and an additional $h - h_0 - 1$ customers must be served with the last customer completing his service exactly at time t.

The probability that ν customers complete their service at station II is given by the ν-fold convolution of the station II service time distribution

$$Y*^{\nu}(\cdot) \tag{13}$$

The probability that the customer currently in service requires τ additional units of service is given by

$$Y(\tau_0 + \tau) \tag{14}$$

where τ_0 is the number of units of service time already completed. Therefore, the probability that after exactly t time units, the station II condition is satisfied is

$$p_{II} = \sum_{\tau_0=0}^{L_2-\tau} Y(\tau_0 + \tau)\, Y^{*(j-j_0)}(t - \tau_0) \tag{15}$$

Similarly, we may derive the probability p_I that the station I conditions are satisfied is the probability that $h - h_0 - 1$ customers are served in $t - \nu$ units of time. This probability is given by

$$p_I = X^{*(h-h_0-1)}(t - \nu). \tag{16}$$

Since the service times at the stations are presumed to be independent, the probability that t units of time are required for the transition is given by

$$\begin{aligned} \psi_1(h_0, j_0, h, j, \nu, \tau, t) &= P(t) = p_I\, p_{II} = \\ X^{*(h-h_0-1)}(t - \nu) &\sum_{\tau_0=0}^{L_2-\tau} Y(\tau_0 + \tau) Y^{*(j-j_0)}(t - \tau_0) \end{aligned} \tag{17}$$

This transition probability is computed for each value of $t = 0, 1, \ldots, T$ where T is the largest possible value of time during which a transition can take place. An upper bound on T is given by:

$$T = \min[L_2(j - j_0),\ \nu + L_1(h - h_0 - 1)]. \tag{18}$$

In the actual computational procedure, T takes on a considerably smaller value because the upper and lower tails of the probability distributions are trimmed which means that beyond a certain point in the distribution, the probabilities are treated as though they were zero. Points in the tails of the distribution which have zero probability are not retained in the storage arrays. This greatly reduces the computation time.

By arguments analogous to the above, we may determine the formulas for each of the quantities described in section 5. We omit details.

(19)
$$\psi_2(h_0,\ j_0,\ h,\ j,\ \nu,\ \tau, t) =$$
$$X^{*(h-h_0)}(t) \sum_{\tau_0=0}^{L_2-\tau} Y(\tau_0 + \tau) Y^{*(j-j_0-1)}(t - \nu - \tau_0)$$

(20)
$$\phi_1(h_0,\ j_0,\ h,\ j,\ \nu,\ \tau,\ t) =$$
$$Y^{*(j-j_0)}(t) \sum_{\tau_0=0}^{L_1-\tau} X(\tau_0 + \tau) X^{*(h-h_0-1)}(t - \nu - \tau_0)$$

(21)
$$\phi_2(h_0,\ j_0,\ h,\ j,\ \nu,\ \tau,\ t) =$$
$$Y^{*(j-j_0-1)}(t - \nu) \sum_{\tau_0=0}^{L_1-\tau} X(\tau_0 + \tau) X^{*(h-h_0)}(t - \tau_0)$$

(22)
$$\phi_1(h_0,\ j_0,\ h,\ j,\ \nu,\ \tau,\ t) =$$
$$Y^{*(j-j_0)}(t) X^{*(h-h_0-1)}(t - \nu)$$

(23)
$$\phi_2(h_0,\ j_0,\ h,\ j,\ \nu,\ \tau,\ t) =$$
$$Y^{*(j-j_0-1)}(t - \nu) X^{*(h-h_0)}(t)$$

In order to make the above formulas valid for all $j \geq j_0$ and $h \geq h_0$, we use the convention that for any probability distribution F, $F^{*0}(\cdot)$ is the probability distribution which assigns probability 1 to the point 0.

6. Computational Procedures for Rectangular Boundary Exits. The phase I processing has been described in Section 4. It amounts to evaluating the formulas for ψ, ϕ, and θ with $\nu = 0$. The processing algorithm is organized so that the processing begins with those lattice points which are likely to have high probability. This point is estimated by beginning with the one closest to the intersection of the phase I boundary with the line through the origin whose slope is

$$(24) \qquad m = \frac{\sum_{i=1}^{L_2} i\, Y_i}{\sum_{i=1}^{L_1} i\, X_i},$$

i.e., the ratio of the means of the Y and X service time distributions.

In later stages of processing, it is necessary to consider an evolution of the process similar to the phase I process. For example, during phase III processing, one computes the conditional distribution along the phase III boundary given that the process passed through a fixed point along the phase II boundary. When the process evolved along this path through the phase II boundary, one of the customers had just completed his service while the other had residual service time remaining. The probability distribution along the phase III boundaries are given by the formulas

$$(25) \quad P[(h, j, \tau, t)] = \sum \phi_i(h_0, j_0, h, j, \nu, \tau, t) P[(h_0, j_0, \nu, t, i)]$$

where (h, j) is a lattice point on the phase III boundary and is the residual service time for customers at station I. The summation is taken over all lattice points (h_0, j_0) along the phase II boundary, ranges over its range $(1, \ldots, L_1$ if $i = 1$ and $1, \ldots, L_2$ if $i = 2)$ and $i = 1, 2$. This formula is valid for probabilities along the upper boundary connecting the points (M, L) and (L, L).

Along the right boundary connecting the points (L, N) and (L, L) the probability is given by

$$(26) \quad P[(h, j, \tau, t)] = \sum \psi_i(h_0, j_0, h, j, \nu, \tau, t) P[(h_0, j_0, \nu, t, i)]$$

where the summation extends over the same indices as in formula (25).

The probability distribution at the corner point (L, L) is given by

$$(27) \quad P[(L, L, 0, t)] = \sum \phi_i(h_0, j_0, L, L, \nu, t) P[(h_0, j_0, \nu, t, i)]$$

where again the summation extends over the indices defined in formula (25). These computations are carried out by a subroutine called SQUARE.

The phase IV processing for total operational time is calculated by the same procedure as for the phase III boundary, except that it is a degenerate case: $j = j_0$ or $h = h_0$ depending on the particular segment of the phase IV boundary.

7. Computational Methods for Triangular Boundaries. The computational strategy and formulas of the preceding section will not suffice for the passage from phase I to phase II calculations because of the triangular shaped regions in which the process must evolve. In the passage from the phase I segment defined by the points (M, 0) and (M, N) to the phase II segment defined by the points (M, N), (L, N), the condition $h \leq j + M$ must be observed by the path. Whenever $h = j + M$, idle time occurs at station I until a customer finishes at station II. This customer then goes to station I for service.

The phase II processing can be reduced to a sequence of rectangular boundary problems in the following way. Consider the lower triangular region. The phase I distributions are computed for the lattice points along the line segment joining the points (M, 0) and M, N). At the point (M, 0) the process must make a transition to the point (M, 1); no other transition is possible because the queue at station I is now empty and a customer at station II must be served. From any point along the line segment joining (M, 1) and (M, N), the process must pass through the points (M, N), (M + 1, N) or one of the points along the segment joining the points (M + 1, 1) and (M + 1, N). The exit distributions at these points are calculated using the same methods as for any rectangular boundary. The rectangle consists of the segments joining the vertices (M, 1), (M, N), (M + 1, N), (M + 1, 1). This rectangle has in common its top side (i.e., the points (M, N) and (M + 1, N)) with the phase II boundary.

The processing then proceeds with the rectangle whose vertices are (M + 1, 2), (M + 1, N), (M + 2, N), (M + 2, 2). In this manner, the calculations are made step by step with the phase I boundary moving in the x direction one step at a time until the lower phase I boundary is mapped onto the phase II boundary.

The upper triangular region consisting of the segments connecting the vertices (0, N), (M, N) and (M, L) is processed in the same manner as the lower triangular region.

8. Tail Trimming and Convolutions. In order to calculate the quantities ψ, ϕ, and θ the convolutions $X^{*j}, j = 0, \ldots, M$ and Y^{*k}, $k = 0, \ldots, N$ are needed. A problem arises in that if M and N are large, a large amount of storage and computational time is needed because the distributions concentrate on a large number of points. Some savings can be effected with negligible loss in accuracy, by trimming the tails of the higher order convolutions. Even though the original service time distribution may concentrate heavily in the tails, the tails of the higher order convolutions will have increasingly smaller probability because of the central limit theorem effect.

An adaptive trimming algorithm was developed to permit trimming both the upper and lower tails of the higher order convolutions. Suppose that the probability distribution is given by

$$p_t, \ t = 0, 1, \ldots, N \tag{28}$$

Then two indices t_ℓ and t_u are chosen corresponding to ε_ℓ and ε_u such that

$$\sum_{t=0}^{t_0} p_t \le \varepsilon_\ell < \sum_{t=0}^{t_0+1} p_t \tag{29}$$

and

$$\sum_{t=t_u}^{N} p_t \le \varepsilon_u < \sum_{t=t_u-1}^{N} p_t. \tag{30}$$

The distribution is then approximated by the new distribution

$$p_t, \ t = t_\ell, t_\ell+1, \ldots, t_u. \tag{31}$$

The tails are ignored. The effect of this trimming is to produce a defective distribution; i.e., one for which the total probability is less than one. The total probability does, however, satisfy the inequality.

$$\sum_{t=t_1}^{t_u} p_t \geq 1 - \varepsilon_\ell - \varepsilon_u = 1 - \varepsilon \tag{32}$$

The effect on the mean and variance of this trimming process can easily be estimated analytically.

Let μ be the mean and μ_T be the mean of the trimmed distributions.

$$|\mu-\mu_T| = \sum_{i=0}^{t_1} ip_i + \sum_{i=t_u}^{N} ip_i \leq \varepsilon_\ell \, t_\ell + \varepsilon_u N \leq N\varepsilon \tag{33}$$

Similarly, let σ be the variance and σ_T be the trimmed variance. Then

$$|\sigma^2 - \sigma_T^2| \leq \sum_{t=0}^{t_1} i^2 p_i + \sum_{t=t_u}^{N} i^2 p_i - (\mu - \mu_T)(\mu + \mu_T) \leq \tag{33}$$

$$N^2 \varepsilon_\ell + N^2 \varepsilon_u - N^2 \varepsilon^2 \leq N^2 \varepsilon$$

The effects of using a trimmed distribution on convolutions is to spread the truncation error over all of the probabilities rather than to concentrate it at a single point. Consider a distribution $F(\cdot)$ and its trimmed version $F_T(\cdot)$. Let G be any other distribution. Let $E(\cdot) = F(\cdot) - F_T(\cdot)$. Then, the total mass of the defective distribution $F_T(\cdot)$ is $| \; 1-\varepsilon$. Consider the convolution $F_T{*}G$. The total mass of $F_T{*}G$ is $1^{\times}(1-\varepsilon)$, the product of the total mass for the two distributions, as may easily be shown. The error term is given by

$$F{*}G - F_T{*}G = E{*}G \tag{34}$$

If the error term E is split into two parts the part trimmed from the upper tail and the part trimmed from the lower tail, then we may consider the support of E*G. Suppose for simplicity that the lower tail is the only one trimmed. Then if G concentrates on the points 0, 1, ..., T_G, the distribution of E*G concentrates on the points 0, 1, ..., $T_G + t_\ell$. The maximum probability associated with a point is bounded by the maximum probability at any point in the G distribution times ε. That this is so may be easily seen by the convolution formula

$$(35) \qquad r_n = \sum_{i=0}^{t_\ell} p_i \, q_{n-i} \leq (\max q_i) \sum_{i=0}^{t_\ell} p_i = \varepsilon \max q_i$$

where q_y is the value of the probability density of the G distribution at the point y and r_n is the probability density of E*G at the point n.

We now consider the savings in storage and the loss in accuracy due to trimming. A specific example is given. For this example, the distribution X is uniform on 10 points and we consider the convolutions X^{*j} for j = 0, 1, ..., 30. In general, the storage required for such a distribution is given by the formula

$$(36) \qquad S_n = 3 + (L_1 - 1)\frac{n(n+1)}{2}$$

where L_1 is the number of points in the distribution for X and n is the number of convolutions. For our example $L_1 = 10$, n = 30, and $S_n = 4188$.

The results are presented in Table I. Six different trimming values ε are given along with the total number of support points for all thirty distributions as well as the number of support points for the thirtieth convolution X^{*30}. The mean of the trimmed version of X^{*30} as well as the total probability mass is given. The trimming procedure used was to trim tails after each convolution ε from each tail of the distribution. Thus, for the fourth distribution, the following was formed:

$$(((X_T * X)_T * X)_T * X)_T * X,$$

where the subscript T denotes trimming.

TABLE I
Effects of Trimming

Trim Value ε	Support Points Total	X^{*30}	Mean	Total Mass
0	4188	275	165.0000	1.00000000
10^{-9}	3282	175	165.0000	.99999997
10^{-8}	3122	163	165.0000	.99999969
10^{-7}	2939	153	165.0003	.99999666
10^{-6}	2734	139	165.0022	.99996895
10^{-5}	2468	123	165.0218	.99964830
10^{-4}	2132	103	165.2055	.99602595

9. Bookkeeping algorithm for Disk Storage. Each of the four phases of the computation results in the computation of the joint distribution of total operational time and residual service time for the customer being served at the time the system reaches a particular lattice point on the boundary. These distributions are stored in a compressed form in an array. However, because of the large amount of space required for the distributions, they are written onto random access disk files as soon as the distribution is generated. Only one lattice point at a time is processed and core space for the distribution at only one point is provided. As the processing proceeds at later stages e.g. from phase I to phase II, it is necessary to retrieve distributions from various points along the boundary. This necessitates using random access files rather than sequential access files.

The standard FORTRAN random access procedures are utilized. However, this access method involves accessing records by record number. The boundary for the various phases of the processing are determined by lattice points which are identified by pairs (h,j). A translation algorithm was devised for translating these pairs into record numbers. The algorithm is further complicated by the fact that processing along any boundary is not necessarily in a logical order along the boundary. Instead, the order of processing is based on probability values in order to take advantage of tail trimming. Another complicating factor is that the coordinate pairs for each boundary assume different values. For instance, along the phase I boundary, the values of h and j lie in the intervals $0 \le h \le M$, $0 \le j \le N$. On the phase II boundary, $M \le h \le M + N$, $N \le j \le M + N$.

Four separate files are utilized for the processing. Unit 1 is used for the phase I boundary; Unit 2 is used for idle time information along the upper triangular boundary and the lower triangular boundary; Unit 3 is used for the phase II boundary. During the computation of the phase II boundary, Unit 4 is used for scratch storage. Unit 4 is also used for the phase III boundary.

Corresponding to each Unit, an array is established with each position in the array corresponding to a lattice point along the boundary. Initially, the array is cleared to zeros and afterward, the entry in the array is the record number of the record which contains the distribution for that particular lattice point.

The particular records for a lattice point are addressed by a pair (j,k) where j = 1 or 2 representing a particular segment of the boundary and k is either the x or y coordinate of the lattice point on the boundary. Thus, for phase I, the points along the segment (h,N) h = 0,1,...,M are addressed by the pair (1,h) while the segment (M,j), j = 0,1,...,N is addressed by (2,j). Similar rules are used for the other units. Table II gives the array index for the lattice points along the phase I boundary, the phase II boundary and the idle time boundary. The example given has five customers initially in each quene.

Table II Indexing System

Unit 1			Unit 2			Unit 3		
h	j	index	h	j	index	h	j	index
5	0	1	5	0	1	5	5	6
5	1	2	6	5	2	6	5	5
5	2	3	7	5	3	7	5	4
5	3	4	8	5	4	8	5	3
5	4	5	9	5	5	9	5	2
5	5	6	10	5	6	10	5	1
4	5	7	0	5	7	5	6	7
3	5	8	1	6	8	5	7	8
2	5	9	2	7	9	5	8	9
1	5	10	3	8	10	5	9	10
0	5	11	4	9	11	5	10	11
			5	10	12			

10. <u>General Results</u>. A set of examples was chosen to illustrate and test the computational ideas developed here. We describe briefly the characteristics of each example. We use the notation (M, N, L_1, L_2) to denote the particular example where M is the number of customers initially at station I and L_1 is the maximum number of service time units a station I customer will need. At station II, there are N customers with a maximum number of service units being L_2.

The following examples were run.

A. (5,5,2,1). The service time at station II required one unit of service with probability one. The other station had a uniform service time distribution.

B. (5,5,2,2). The service time distribution at each of the stations was uniform; i.e. with probability ½ one unit was required and with probability ½ two units were needed.

C. (5,5,2,3). The service time distribution at station I is uniform; while at station II, $p_1 = .5$, $p_2 = .25$, $p_3 = .25$

D. (5,5,3,2). This example is the same as example B.

E. (5,10,2,2). The service time distribution is uniform at each station.

F. (10,5,2,2). Again the service time distribution is uniform at each station.

G. (10,10,2,2). The service time distribution is uniform at each station.

H. (15,15,4,4). The service time distribution is uniform at each station.

Example H is the largest complete computation which was made. Phase I computations were made for an example with 30 customers at each station. The main purpose of these examples was to serve as a check on the computations, test the program for correct processing of asymmetric examples and estimate processing time.

Table III lists the total operational time distribution for the (5,5,2,1) example. The distribution is not binomial, because there is idle time at station II during the phase II processing.

Table III

The distribution of total operational time for the (5,5,2,1) system.

Time	Probability	Time	Probability
10	.00097656	16	.17578125
11	.00976562	17	.10742187
12	.05371094	18	.04394531
13	.14648437	19	.00976562
14	.22460937	20	.00097656
15	.22656250		

The complete results for the (15,15,4,4) example are given for the untrimmed case and the trimmed case. Table IV gives the exit probabilities at the phase I boundary. Three separate calculations are given to show the effects of trimming on these probabilities. In the untrimmed case, no trimming is used, while in the second case, the probabilities are trimmed at $\epsilon = 10$ and in the third case, the probabilities were trimmed at 10^{-6}.

Position H	Position K	Untrimmed PROBABILITY	$\varepsilon = 10^{-8}$ PROBABILITY	$\varepsilon = 10^{-6}$ PROBABILITY
15	0	0.0		
15	1	0.0		
15	2	0.0		
15	3	0.00000000		
15	4	0.00000001	0.00000001	
15	5	0.00000091	0.00000090	0.00000074
15	6	0.00002313	0.00002313	0.00002288
15	7	0.00026704	0.00026704	0.00026680
15	8	0.00175876	0.00175875	0.00175851
15	9	0.00753842	0.00753842	0.00753822
15	10	0.02287019	0.02287019	0.02287004
15	11	0.05202916	0.05202916	0.05202888
15	12	0.09255192	0.09255192	0.09255180
15	13	0.13285647	0.13285646	0.13285630
15	14	0.15771523	0.15771523	0.15771513
15	15	0.06477752	0.06477752	0.06477745
0	15	0.0	0.00000001	0.00000074
1	15	0.0	0.00000090	0.00002288
2	15	0.0	0.00002313	0.00026680
3	15	0.00000000	0.00026704	0.00175851
4	15	0.00000001	0.00175875	0.00753822
5	15	0.00000091	0.00753842	0.02287004
6	15	0.00002313	0.02287019	0.05202888
7	15	0.00026704	0.05202916	0.09255180
8	15	0.00175876	0.09255192	0.13285630
9	15	0.00753842	0.13285646	0.15771513
10	15	0.02287019	0.15771523	
11	15	0.05202916		
12	15	0.09255192		
13	15	0.13285647		
14	15	0.15771523		

Table IV

Phase I exit probabilities

Trimming is understood to mean that in distributions, tail probabilities which are below ε (= 10^{-8} and 10^{-6} respectively) are truncated to zero and these points are deleted from the distribution. Moreover, any conditional distribution at any of the lattice points in the hexagon H whose total probability is less than ε are completely ignored.

Table Va gives the total operational time distribution for the untrimmed and each of the two trimmed examples. Table Vb gives a summary of the computational time for each phase of the computation. Table Vc gives a list of the amount of probability lost at each stage due to trimming. Table VI gives the probability distribution at the phase III boundary. These are closely related to the idle time distributions.

Table Va
Operational Time Distribution

Time	$\varepsilon = 0$ Probability	$\varepsilon = 10^{-8}$ Probability	$\varepsilon = 10^{-6}$ Probability
55	0.0	0.0	0.0
56	0.000001	0.000001	0.000001
57	0.000003	0.000002	0.000002
58	0.000007	0.000006	0.000006
59	0.000019	0.000016	0.000016
60	0.000046	0.000041	0.000041
61	0.000108	0.000096	0.000097
62	0.000239	0.000213	0.000215
63	0.000500	0.000448	0.000453
64	0.000990	0.000893	0.000903
65	0.001859	0.001687	0.001705
66	0.003312	0.003024	0.003055
67	0.005608	0.005148	0.005202
68	0.009029	0.008334	0.008421
69	0.013838	0.012837	0.012971
70	0.020202	0.018832	0.019030
71	0.028120	0.026334	0.026614
72	0.037350	0.035130	0.035507
73	0.047378	0.044741	0.045228
74	0.057442	0.054445	0.055054
75	0.066623	0.063358	0.064085
76	0.073981	0.070563	0.071400

77	0.078724	0.075275	0.076204
78	0.080348	0.076986	0.077980
79	0.078726	0.075549	0.076576
80	0.074123	0.071203	0.072228
81	0.067126	0.064509	0.065499
82	0.058526	0.056233	0.057158
83	0.049174	0.047207	0.048043
84	0.039852	0.038200	0.038931
85	0.031180	0.029820	0.030439
86	0.023569	0.022475	0.022982
87	0.017225	0.016366	0.016768
88	0.012178	0.011521	0.011829
89	0.008333	0.007843	0.008072
90	0.005520	0.005166	0.005331
91	0.003541	0.003292	0.003407
92	0.002199	0.002030	0.002107
93	0.001322	0.001211	0.001261
94	0.000769	0.000699	0.000730
95	0.000433	0.000390	0.000409
96	0.000235	0.000210	0.000221
97	0.000123	0.000109	0.000115
98	0.000063	0.000055	0.000058
99	0.000031	0.000026	0.000028
100	0.000014	0.000012	0.000013
101	0.000006	0.000005	0.000006
102	0.000003	0.000002	0.000003
103	0.000001	0.000001	0.000001

Table Vb

computation time

Phase	$\varepsilon = 0$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-6}$
I	1.016	.958	.923
II	38.341	27.723	21.526
III	157.054	110.413	96.315
IV	1.376	1.037	.926
Total	197.787	140.131	119.690

Table Vc

Trimming Effects

	Probability lost		
Phase	$\varepsilon = 0$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-6}$
I	0	3×10^{-8}	4×10^{-6}
II	0	3×10^{-2}	1×10^{-2}
III	0	2×10^{-2}	2×10^{-2}
IV	0	0	0

The total number of points on which the final operational time distribution resides is 95 for $\varepsilon = 0$, 68 for $\varepsilon = 10^{-8}$ and 62 for $\varepsilon = 10^{-6}$. A strange pattern exists between the ε values and the probability mass lost due to trimming The case $\varepsilon = 10^{-8}$ seems to have lost more probability than the case $\varepsilon = 10^{-6}$. Also, the total probability lost, particularly in the later stages of the computation, is relatively large. A smaller ε value in these late stages of the computations would probably remedy the situation.

Table VI

Phase III boundary probabilities

H	K	$\varepsilon = 0$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-6}$
30	15	0.00000047		
30	16	0.00000353	0.00000003	0.00000093
30	17	0.00001917	0.00000217	0.00001085
30	18	0.00008556	0.00002586	0.00005624
30	19	0.00031799	0.00015313	0.00022866
30	20	0.00100092	0.00060582	0.00076979
30	21	0.00271011	0.00184516	0.00217903
30	22	0.00640707	0.00472150	0.00536664
30	23	0.01341559	0.01062907	0.01178075
30	24	0.02516235	0.02135680	0.02314250
30	25	0.04252185	0.03824821	0.04054072
30	26	0.06477119	0.06079866	0.06320016
30	27	0.08875024	0.08566056	0.08772284
30	28	0.10925278	0.10721851	0.10868896
30	29	0.12099172	0.11984512	0.12072813
30	30	0.04917891	0.02668752	0.02668738
15	30	0.00000047	0.00000087	0.00000275
16	30	0.00000353	0.00001266	0.00001852
17	30	0.00001917	0.00007661	0.00008490
18	30	0.00008556	0.00031027	0.00031719
19	30	0.00031799	0.00099642	0.00100009
20	30	0.00100092	0.00270824	0.00270923
21	30	0.00271011	0.00640649	0.00640630
22	30	0.00640707	0.01341545	0.01341467
23	30	0.01341559	0.02516230	0.02516173
24	30	0.02516235	0.04252140	0.04252101
25	30	0.04252185	0.06476609	0.06477032
26	30	0.06477119	0.08871662	0.08874692
27	30	0.08875024	0.10910787	0.10923414
28	30	0.10925278	0.12054744	0.12091419
29	30	0.12099172		

11. Waiting time for the ith customer. The problem of finding the waiting time distribution for the ith customer initially at station I (or station II) is easily solved using the method and programs described in the previous sections. Suppose that we wish to find the waiting time distribution of the customer who is initially customer i at station I. This customer i must be served by both stations I and II. All of the customers in line behind customer i will necessarily be served after he is. As far as customer i is concerned, these customers have no relevance. Thus one can consider a reduced queueing system which has initially i customers at station I and N customers at station II.

In the reduced queueing system, customer i is now the last customer at station I. His total waiting time is that required for him to clear station II. We may describe the waiting time problem in terms of the random walk representation pictured in figure 1. In this representation, the last customer initially at station I clears station II when the sojourn path beginning at the origin reaches the upper boundary, i.e. the segment connecting the lattice points (i,L) to (L,L). At this time, station I may still be serving, but station II's service is complete.

In the notation of the preceding sections, a formula may be written for the waiting time distribution for the ith customer as follows.

$$p_t = \sum_{k=1}^{N} \sum_{\tau=1}^{L_I} P(i + k, L, \tau, t), \quad t = 0,1,.... \tag{37}$$

where P is the probability distribution at the lattice point (i + k, L) along the upper boundary, and τ is the number of service time units remaining for the customer at station I.

12. Idle time distribution. There are two types of idle time at each of the serving stations. At station I, idle time occurs at the lower boundary in figure 1 which corresponds to the line at station I being empty, but there remain customers at station II who must also be served by station I. Idle time also occurs at the right boundary in figure 1. This corresponds to the completion of service at station I for all customers in the system, but there yet remain customers at station I who require service. Similarly at station II idle time of each type occurs.

The idle time distribution of the second type is relatively easy to obtain once the probability distribution for the phase III boundary is obtained. This distribution is given for station I by

$$p(t) = \sum_{j=N}^{L} \sum_{\tau=1}^{L_2} y^{*(L-j)} (t - \tau) \, P \, [j, L, \tau]$$

where $P \, [j, L, \tau]$ is the probability of passage to the lattice point (j,L) with τ units of residual service time at station II. This probability is easily obtained from the phase III boundary probabilities by summing out the time parameter. A similar formula is available for the station II idle time distribution.

The other type of idle time distribution which occurs at the lower boundary and at the left boundary. This probability calculation is not as readily available as a by product of the total operational time distribution. The programs and algorithms could be modified to keep tract of the idle time of this type as well as the total operation time. This procedure would amount to keeping tract of the joint distribution of the idle time as well as the total operational time. This would require a considerable increase in the amount of storage needed. Another alternative is to modify the algorithm to keep tract only of the idle time at the triangular boundaries.

II. Comparison With Simulation Methods

A simulation study was made in order to assess the exact computational algorithms for correctness. An additional benefit was to determine how much accuracy in the operational time distribution could be obtained by simulation methods. In this section, we discuss the random number generation methods and testing, the simulation algorithm, and a collection of results.

1. Random number generation. Three methods for generating random numbers uniformly distributed over the unit interval were considered. They were: (a) linear congruential, (b) random numbers from an adapted form of the Rand Corporation's million random digits, and (c) a mixture of the two.

The linear congruential generator used was the usual one in which the recurrence formula

$$X_{n+1} = aX_n \mod(b)$$

is used to generate the random sequence. In this problem, the value of a used was 32771 and the value of b was 2^{32}.

The random numbers from the Rand Corporation's million random digits were converted to twenty-four bit binary integers and used as random numbers distributed uniformly on the unit interval by a suitable positioning of the binary point.

One of the criticisms of linear congruential generators is the possibility of serial correlations and lack of randomness in runs. The purely random numbers suffer from the relatively small number of them. In our case the total number of such numbers was about 138,000.

A method suggested for solving these difficulties is to mix the random numbers with pseudo random numbers in the following way. Generate a sequence of pseudo random numbers $\{X_n\}$ as described above as well as a sequence of random numbers $\{Y_n\}$ which have the property that $Y_n + 138{,}000 = Y_n$, i.e. use the first 138,000 random numbers and after that reuse the original sequence. The mixed random number Z_n is obtained by $Z_n = X_n \circledast Y_n$ where the symbol $\circledast$ means that Z_n is obtained using the exclusive or on the digits of the binary representation of X_n and Y_n to obtain the binary digits of Z_n. The advantage of this mixing is that the sequence $\{Z_n\}$ enjoys the randomness properties of the random numbers and also the advantages of the pseudo random sequence.

For each of the methods of random number generation described above, the following tests were made: (a) distribution test, (b) serial correlation test, and (c) runs test. For each test, 10,000 random numbers were generated. The distribution test consisted of partitioning the unit interval into 1000 subintervals of equal length, obtaining the empirical frequency count and applying a chi-square goodness of fit test. The serial correlation test consisted of computing the one step serial correlation for the pairs (Z_{Y+1}, Z_Y), $Y = 1,2,\ldots, n-1$. The runs test is described in [11] p. 60. In this particular test, we counted runs of length 1,2,3,4,5 and greater than or equal to 6. The observed number of runs was compared with the expected and a chi-square value was calculated. In all cases, the chi-square values were close to the expected values as well as the overall mean and variance. The results are given in table VII. They suggest that any of the methods of random number generation are acceptable.

Table VII

Simulation Results

	pseudo random	random	mixed
overall mean	.5030	.5008	.4978
overall variance	.0838	.0831	.0837
serial correlation	.008644	-.001877	-.004844
distribution chi square (999df)	1036.6	1028.4	1029.4
run length chi square (6df)	5.471	7.136	4.092

2. Simulation Algorithm. A relatively straight forward simulation algorithm was used. For a given queue setup, a set of service times was generated for each station, according to the service time distribution for that station, and for each of the M + N customers to be served by that particular station. The successive service times were independent from customer to customer as well as from station to station. Once the service times were generated, an algorithm determined the overall operational time for the system. The total service time was not merely the larger of the sum of the two service times for each station, because of the possibility of idle time at one station while it awaits customers from the other. This consideration was easily accomodated by the algorithm. We omit further details.

3. Comparison of Numerical Results with Simulation Results. We now list for comparison some simulation results with exact numerical computations. A

chi square goodness of fit test was made to compare the numerical results with the simulation results for the total operational time distribution of examples B through H of section I. 10. The chi square values are given in table VIII. In table IX, the total operational time distribution for the (5, 5, 2, 3) example is given as obtained by simulation. The number of simulations was 10,000 in each case.

Table VIII

Comparison of Simulation With Exact Calculations

Example	Chi square	df	True Distribution Mean	True Distribution Variance	Empirical Distribution Mean	Empirical Distribution Variance
(5, 5, 2, 2)	31.34	9	15.84	1.67	15.90	1.74
(5, 5, 2, 3)	93.10	15	17.75	4.82	17.88	5.34
(5, 5, 3, 2)	51.07	15	17.75	4.82	17.86	5.09
(10, 5, 2, 2)	14.49	11	23.62	2.66	23.69	2.57
(10,10, 2, 2)	24.92	13	31.20	3.34	31.27	3.36
(15,15, 4, 4)	45.44	33	78.27	24.79	78.47	25.62

The general results show that as far as the mean and variance are concerned, the simulation produces close, but not accurate results. However, as far as the overall distribution is concerned, the good ness of fit is poor, especially when there are a small number of customers in the system. An analysis of the results shows that the poor fit is in the tails of the distribution. An extreme example is given in Table IX for the (5, 5, 2, 3) example which gave the poorest fit of all the examples run.

Table IX

Goodness of Fit Results

Operational Time	Exact Probability	Frequency Count	Expected Frequency	Contribution to X^2
10	0.00000095	0	0.0	0.01
11	0.00006199	0	0.6	0.62
12	0.00110626	10	11.1	0.10
13	0.00869751	105	87.0	3.74
14	0.03723145	378	372.3	0.09
15	0.09591246	923	959.1	1.36
16	0.16157508	1517	1615.8	6.04
17	0.19060600	1869	1906.1	0.72
18	0.17199892	1621	1720.0	5.70
19	0.13040411	1259	1304.0	1.56
20	0.08936733	957	893.7	4.49
21	0.05579908	614	558.0	5.62
22	0.03127172	378	312.7	13.63
23	0.01547158	227	154.7	33.77
24	0.00676496	97	67.6	12.73
25	0.00257871	35	25.8	3.29
26	0.00085335	7	8.5	0.28
27	0.00023588	2	2.4	0.05
28	0.00005317	1	0.5	0.41
29	0.00000861	0	0.1	0.09
30	0.00000090	0	0.0	0.01

III. General Conclusions

The results of this study have shown that it is possible to analyze complex queueing models of the type discussed in this report using discrete time methods together with modern computational facilities. Heretofore, analytic methods have been able to solve only the simplest cases of queues having exponential service times. The analytic methods rely heavily on the memoryless property of the exponential distribution. Using discrete time methods, proposed by Neuts [10] and others, the queueing problem is completely solvable with arbitrary service time distributions. The results reported here show that queues with fifteen customers initially at each station and a service time distribution, a very small amount of computer time is needed. With service time distributions concentrating on a larger number of points, say sixteen points, for each station, the computation time would increase by a factor of sixteen or less depending on the effects of trimming. The amount of computer time needed would still remain within reasonable bounds -- about thirty minutes based on extrapolations from out results. Still larger problems could be solved, but refinements of the algorithms would be needed. In particular, the use of the Fast Fourier Transform algorithm for convolutions of distributions concentrating on a large number of points would considerably decrease processing time. For the small examples studied in this report, the use of the Fast Fourier Transform would not be efficient, since convolution via this transform method is faster than the direct methods only when the particular distributions involved concentrate on more than 128 points.

The commonly used simulation methods for solving queueing problems of this type are not as useful for obtaining the operational time probability distribution as their common use would indicate. Our experience indicates that the mean and variance can be obtained reasonably well if one is satisfied with two digit accuracy. An additional digit or so is possible by increasing the number of simulations. However, the frequency counts do not give accurate estimates for the probability density. The problem is more serious for queues with smaller numbers of customers rather than larger ones. The chi square tests generally fail until there are ten customers in the queue. This phenomena is probably due to a tendency for the operational time distribution to exhibit a limiting behavior as the number of customers at a station increases.

One of the major drawbacks to this method of analysis is the rather large amount of complex computer programming which is required. The overall amount of t time involved is certainly larger than that required for a simulation, but probably not much more than would be required via analytic methods.

A further aspect of this problem which needs to be analyzed is the approximation procedure which would be required to model an exponential service time distribution by a discrete time distribution, say a truncated geometric distribution.

Related Literature

1. Emmons, Hamilton (1973), The two machine job shop with exponential processing times, Technical Report.

2. Feller, W. (1957), An Introduction to Probability Theory and Its Applications, Vol. I, Wiley, N.Y.

3. Gordon, W.J. and Newell, G.F. (1967), Closed queueing systems with exponential servers, Operations Res., 15, pp 254-265.

4. Heimann, David and Neuts, Marcel F. (1974), The single server queue in discrete time - numerical analysis IV, Nav. Res. Log. Quart. 20, pp. 753-766.

5. Jackson, J.R. (1963), Johshop like queueing systems, Management Sci. 10, pp. 131-142.

6. Klimko, E. M. and Neuts, Marcel F. (1973), The single server in discrete time - numerical analysis II, Nav. Res. Log. Quart. 20, pp. 305-319.

7. Klimko, E. M. and Neuts, Marcel F. (1973), The single server in discrete time - numerical analysis III, Nav. Res. Log. Quart. 20, pp. 557-567.

8. Klimko, E. M. (1977), Discrete time analysis of a shutdown queueing system, (in preparation).

9. Milch, P.R. and Waggoner, M. H. (1970), A random walk approach to a shutdown queueing system, Siam J. Appl. Math., 19, pp. 103-115.

10. Neuts, Marcel F. (1973), The single server queue in discrete time - numerical analysis I, Nav. Res. Log. Quart. 20, pp. 297-304.

11. Knuth, D. F. (1969), The Art of Computer Programming, Vol. II, Seminumerical Algorithms. Addison Wesley, Reading, Mass.